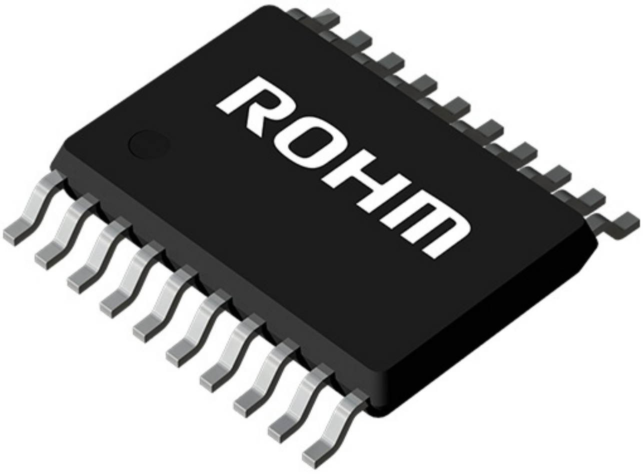
Contents [hide]

ROHM

ROHM ML62Q203x LogiCoA RMOS Microcontroller for Power Supply Control



Specifications

- **Product Name**: LogiCoATM Power Solutions
- **Manufacturer**: ROHM Co., Ltd.
- **Communication Protocol**: UART
- **Operating System Compatibility**: Windows 10 64-bit
- **Required Software**: Microsoft Excel 64-bit version
- **Power Supply Output**: 12V, 1A or higher

Introduction

The microcontroller for Power Supply Control ML62Q203x/ML62Q204x (hereinafter referred to as ML62Q20xx group) has two communication interfaces of UART/I2C. RMOS (Real-time Micro Operating System) provided by us uses UART to create a frame configuration and a packet configuration in advance and provide it as a software that enables communication control using Excel VBA. Communications between the MCU and PC can be easily performed by writing RMOS and setting commands arbitrarily in Excel VBA. The user can also easily create a GUI to control communications. Another benefit is that it implements a communications protocol that can communicate with up to 32 power supplies for a single PC by software processing RMOS. This application note introduces the protocol for serial communication using UART, explanations of communication functions (such as how to append to commands), and shows how to use the communication GUI development environment. The communication described here uses a buck DCDC converter (hereafter buck converter) EVK, LogiCoA001-EVK-001 (hereafter buck converter EVK).

Usage environment

Software and Equipment

Communication using RMOS described in this manual is performed using the following.

1. Buck converter EVK
2. USB-UART convert module: FT234x FTDI (mounted on 1)
3. Windows PC(Windows10 64bit)
4. On-chip emulator “EASE1000 V2”
5. RMOS project file(file to be read into LEXIDE-Ω and used)
6. Microsoft Excel 64-bit version (Operation check: Microsoft 365 MSO, Office 365 MSO)

7. Excel file “RMOS_CommunicationControl_LogiCoA001-EVK-001.xls”
8. Regulated DC power supply (Performance: Output-voltage 12V, Output current 1A or higher)
9. Digital multimeter

Figure 1-1 shows a simplified diagram for communication. When checking the operation, disconnect the on-chip emulator and connect the measuring device to the I/O pin of the buck converter EVK.

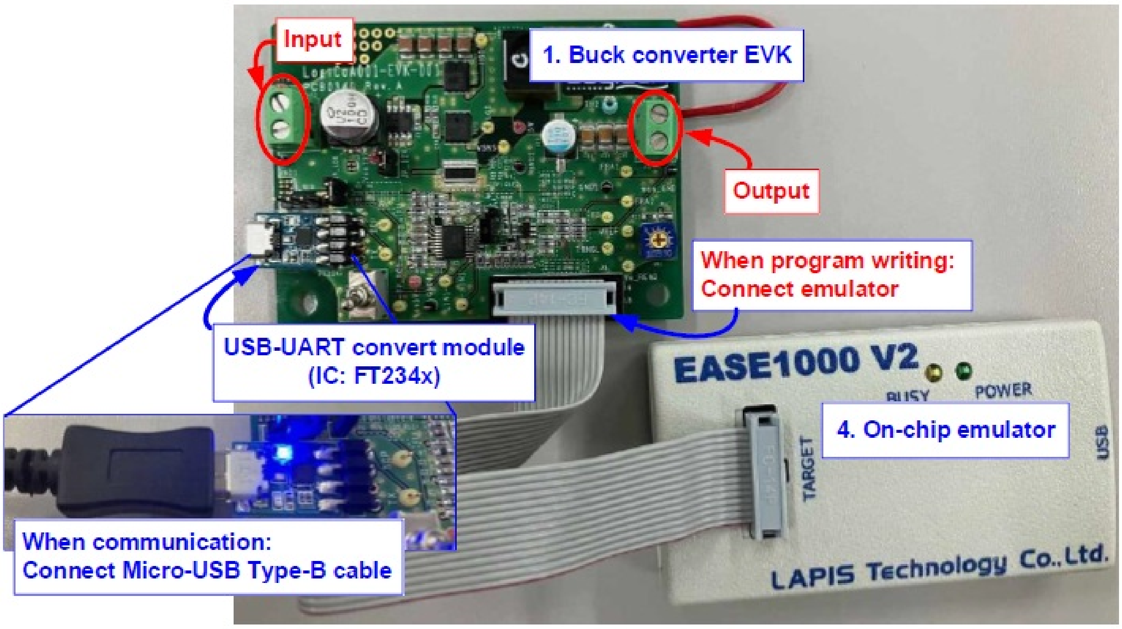


Figure 1-1. Connection diagram for communication

For details on how to connect the Figure 1-1 buck converter EVK and on-chip emulator, and how to write RMOS codes, refer to RMOS application note [1].

Files in RMOS folders are required for communication control

Table 1-1 describes the RMOS file necessary for communication control described in this manual. How to change the file name of the table and how to check the operation after changing the description are explained in the later chapters.

Table 1-1. Files in RMOS folders are required for communication control

No.	Folder name	File name	Description	Changeable part

1	00_System	00S93_Func_Select.s	Function setting of OS during program debugging	Communication mode
2	10_Setting	10S01_Parameter_Init.s	Set the initial value of power supply operation parameters	Communication address
3	“	※10S70_UART_Set.s	UART assignment and operation setting	※No need to change
4	30_Info_module	30I01_InfoCMD_Exec.asm	Write the program of communication command	Add program of additional command
5	“	30I11_InfoCMD_Table_def.s	Communication command assignment	Assign labels for additional commands
6	90_Header	※93H010_InfoTxRx_Header.s	Describes a program for processing transmission and reception data	※No need to change

Communication protocol

This chapter describes the configuration of serial communication using UART set in RMOS. The hardware used to communicate between the microcontroller and PC is UART., RMOS has a communication protocol that consists of 6 frames per packet by software processing. This packet configuration enables one PC to perform communication control of 32 power supplies and communication from PC to the power supply circuitry using several commands. Table 2-1 indicates UART hardware configuration and Table 2-2 indicates RMOS communication frame configuration. All the initialization settings in the table are set in the “10S70_UART_Set.s.s” file. 2nd bit of Table 2-2 is identified by the frame identification bit, the first frame (Frm0) is identified by “1”, and the remaining frames (Frm1-Frm5) are identified by “0”.

Table 2-1. UART hardware configuration

	Setting
Baud rate	9600bps
Data length	8bit
communication direction	LSB first
Parity	Use, Even
Stop bit	1bit

Table 2-2. Frame configuration

bit	Assignment
1	Start
2	Data (Identification)
3	Data
4	Data
5	Data
6	Data
7	Data
8	Data
9	Data
10	Parity
11	Stop

Figure 2-1 packet configuration is described below. One packet of transmission/reception consists of 6 frames (= total 42 bits in length): address length (5 bits) + decision bit (1 bit) + data length (32 bits) + checksum length (4 bits). All the software processes that process 6 frames into one packet are described in the “93H010_InfoTxRx_Header.s.s” file. There are two transmission methods: TX32 and TX16 (the communication explained in this document is set by TX16). TX32 can transmit the data of 32-bit length up to 7th-38th bit of the packet to the microcontroller, but the number of commands is limited to one. On the other hand, TX16 has 8bit lengths for command groups (CmdGr) and command numbers (CmdNo) from the 7th to 22nd bit, and up to 65536 types of commands can be used in combination (For the communication commands of the buck converter, areas from 0-15 for CmdGr and from 0-31 for CmdNo can be used.). The data to be transmitted to the microcontroller is 16 bits in length from the 23rd to the 38th bit. ADR (1st-5th bit) of Frm0 is assigned to the connected devices, and up to 32 power supplies can be connected simultaneously. 6bit of Frm0 is determined by TX32 and TX16, where “1” is TX32 and “0” is TX16. The chksum(39th-42nd bit) is an error detection and judgment bit. This bit is used to detect and judge errors in the communication data of all 6 frames. In addition, the receiver RX returns the data of 32 bits in response to the transmitted command.

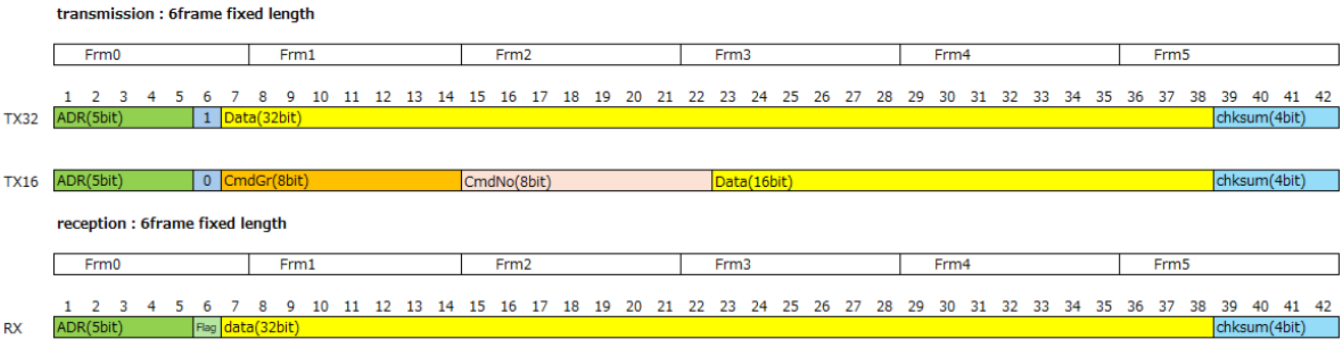


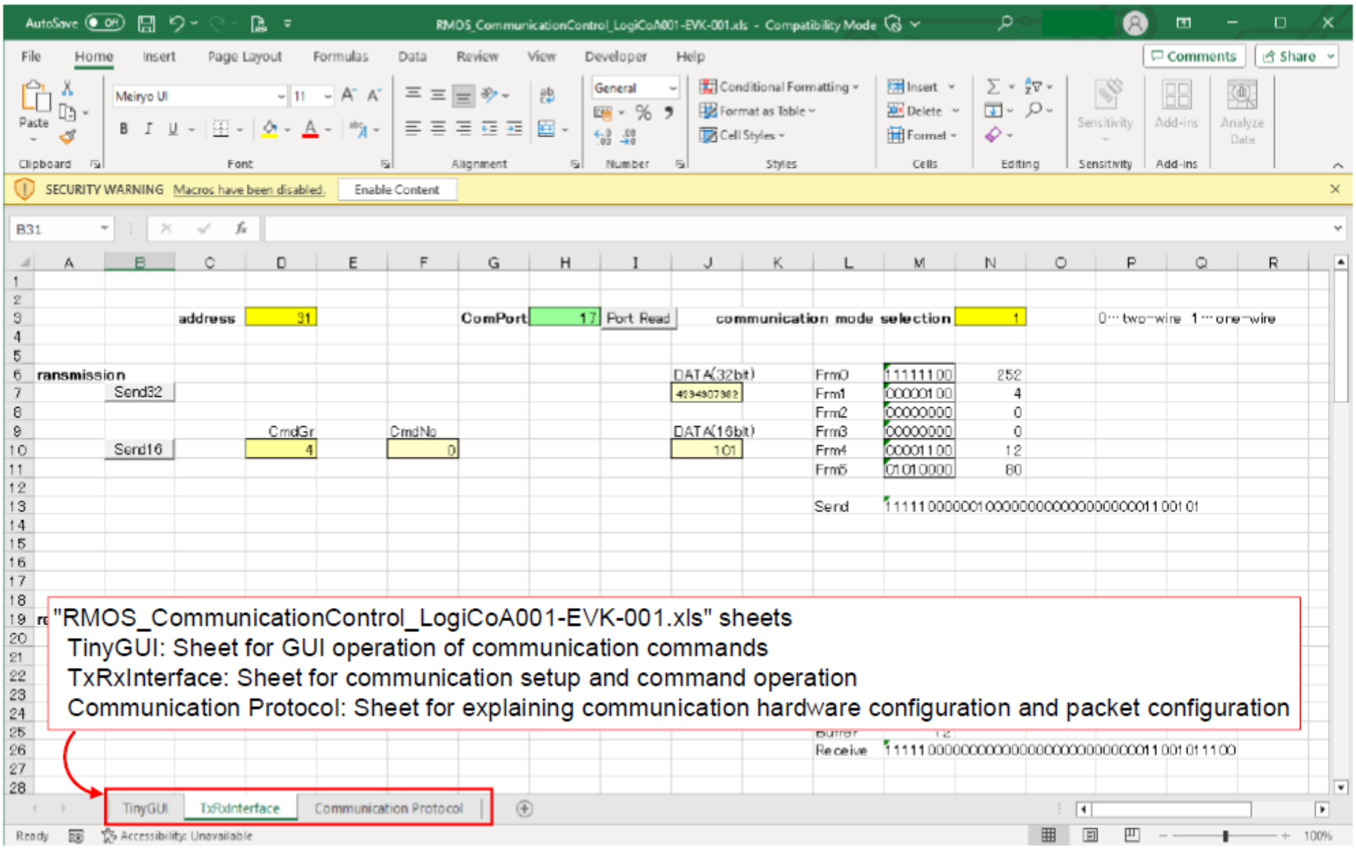
Figure 2-1. Packet configuration

Communication method

Initialization Settings of Excel VBA and RMOS

Excel VBA setting

This section describes the initialization of Excel files required for communication between the PC and the microcontroller. First, start the “RMOS_CommunicationControl_LogiCoA001-EVK-001.xls” file downloaded from ROHM website. The operating sheets are described in later chapters.



"RMOS_CommunicationControl_LogiCoA001-EVK-001.xls" sheets

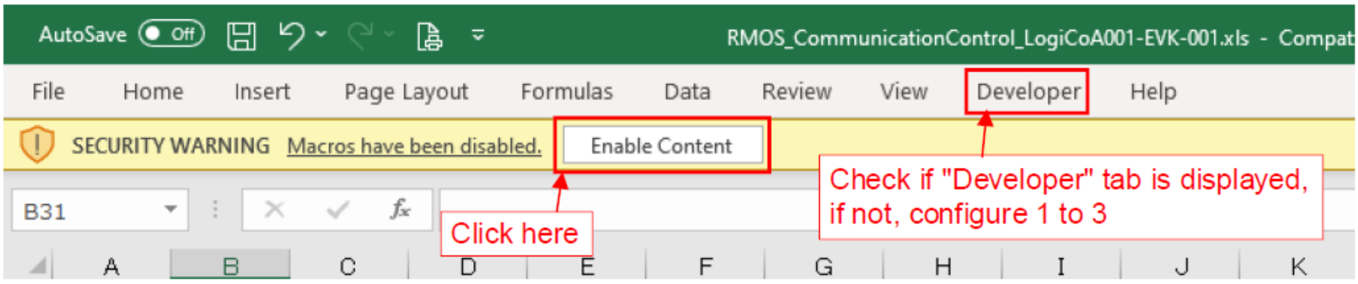
TinyGUI: Sheet for GUI operation of communication commands

TxRxInterface: Sheet for communication setup and command operation

Communication Protocol: Sheet for explaining communication hardware configuration and packet configuration

Figure 3-1. Startup window for Excel file

Next, check whether Excel is set up to use VBA macros. If “Development” tab is displayed in Excel (Figure 3-2), VBA macro can be executed.



AutoSave Off
RMOS_CommunicationControl_LogiCoA001-EVK-001.xls - Compat
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Help
SECURITY WARNING Macros have been disabled.
Enable Content
B31
Click here

Check if "Developer" tab is displayed, if not, configure 1 to 3

Figure 3-2. Top page of Excel

If “Developer” tab is not displayed, follow the step below to configure the settings.

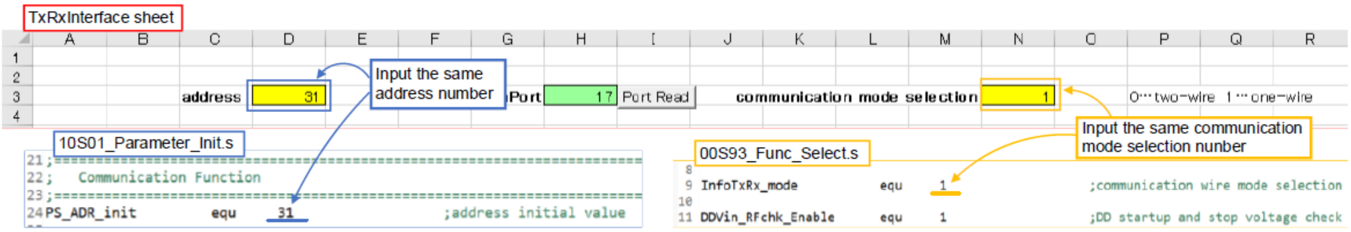
1. Click “File” tab > “Options” > “Customize Ribbon”.
2. Check the “Developer” box in the “Customize the Ribbon” list.
3. Click “OK”.
4. Check the window that the “Developer” tab is displayed.

Also, the first time the user opens an Excel file, “Macro have been disabled” message is displayed. Click on “Enable Content” button to improve. Once the user does this in the first use, the user will not receive the warning message the next time the user launches

Excel file.

Setting of the communication address and communication mode

Check the serial communication mode of the UART and the microcontroller address setting. Configure the settings so that Excel filename matches RMOS description as shown in Figure 3-3.



TxRxInterface sheet

address 31 | Port 17 | Port Read | communication mode selection 1 | 0… two-wire 1 … one-wire

10S01_Parameter_Init.s

```
;==========================================================================
;    Communication Function
;==========================================================================
PS_ADR_init          equ     31                     ;address initial value
```

00S93_Func_Select.s

```
8
9  InfoTxRx_mode          equ     1          ;communication wire mode selection
10
11 DDVin_RFchk_Enable     equ     1          ;DD startup and stop voltage check
```

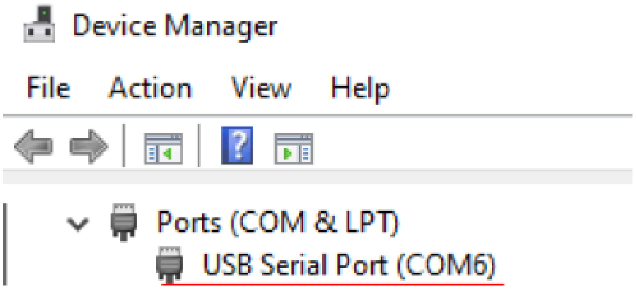
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
1
2
3
4
Input the same address number
Input the same communication mode selection nunber

Figure 3-3. "00S93_Func_Select.s" file and Excel function setting

- **Blue:** The value in D3 cell is the address of the microcontroller. In the provided RMOS, PS_ADR_init “31” in “10S01_Parameter_Init_Init_ Init.s” file is the address of the microcontroller. There is no need to change the address in this manual. However, when connecting more than one power supply to PC for communication control, set the address of each power supply on RMOS and specify the address of the target power supply in Excel.
- **Yellow:** N3 cell is used to change UART communication mode. The communication in LogiCoATM power solution has two connection methods: two wires and one wire. “InfoTxRx_mode” on line 9 of “00S93_Func_Select.s” file can be set to “0” for two wires and “1” for one wire connection mode. The buck converter EVK described in this manual is designed for one wire communication.

Checking and setting COM port number

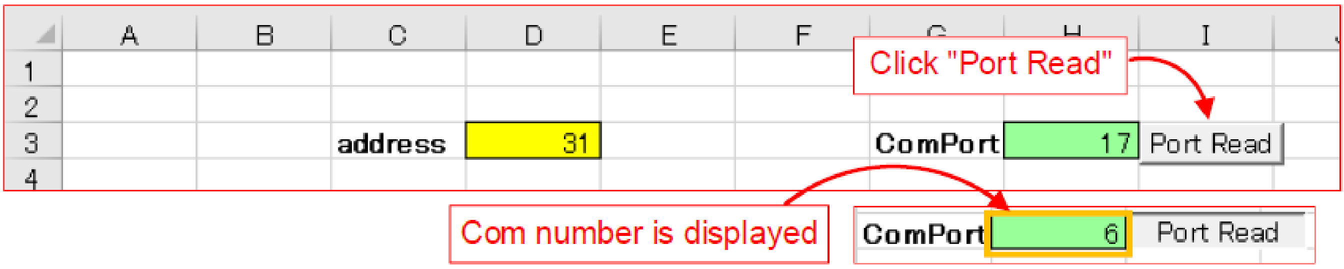
Please acquire COM port number of TxRxInterface sheet. COM port is a type of communication port for connecting PC to an external device and is used for serial communication. When USB cable is connected to the USB-UART convert module, Figure 3-4 will show “USB Serial Port (COM6)” in Windows Device Manager (Driver will install automatically if connected to a network). The number “COM*” is COM port number. The number varies depending on PC and the USB-UART convert module.



Ports (COM & LPT)
USB Serial Port (COM6)

Figure 3-4. Device manager window

NOTE: Even if the USB cable is connected, the driver may not be automatically installed and the Figure 3-4 window may not display. In this case, the driver must be downloaded and installed from the FTDI website. This COM number must be specified when communicating with Excel file. In the TxRxInterface sheet, VBA macro automatically acquires the COM port number of USB module. When click “Port Read” on Figure 3-5, the number is displayed in H3 cell.



	A	B	C	D	E	F	G	H	I
1									
2									
3			address	31			ComPort	17	Port Read
4									

Click "Port Read"

Com number is displayed

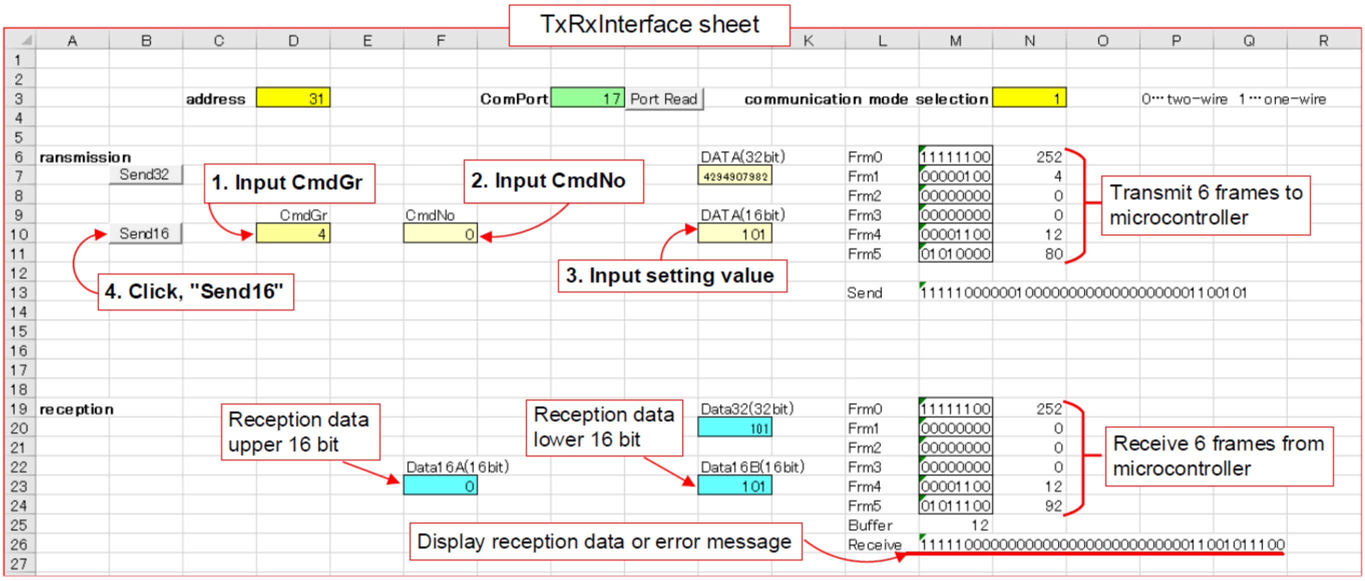
ComPort 6 Port Read

Figure 3-5. TxRxInterface sheet port read key

If communication between the PC and the MCU fails, first check that the COM port number has been acquired correctly. If more than one USB is connected, the port number may not be acquired from the Excel file. To resolve this, click “Port Read” again, or check COM number in Device Manager and input the COM number in the H3 cell.

How to use the TxRxInterface Sheet

Perform produces 1 to 4 below to perform communication using the TxRxInterface sheet.


TxRxInterface sheet

address 31

ComPort 17

communication mode selection 1

0… two-wire 1… one-wire

ransmission

CmdGr 4

CmdNo 0

DATA(32bit) 4294907982

DATA(16bit) 101

Frm0	11111100	252
Frm1	00000100	4
Frm2	00000000	0
Frm3	00000000	0
Frm4	00001100	12
Frm5	01010000	80

Send 1111100000010000000000000000000001100101

Transmit 6 frames to microcontroller

1. Input CmdGr

2. Input CmdNo

3. Input setting value

4. Click, "Send16"

reception

Data32(32bit) 101

Data16A(16bit) 0

Data16B(16bit) 101

Frm0	11111100	252
Frm1	00000000	0
Frm2	00000000	0
Frm3	00000000	0
Frm4	00001100	12
Frm5	01011100	92

Buffer 12

Receive 11111000000000000000000000000000011001011100

Reception data upper 16 bit

Reception data lower 16 bit

Receive 6 frames from microcontroller

Display reception data or error message

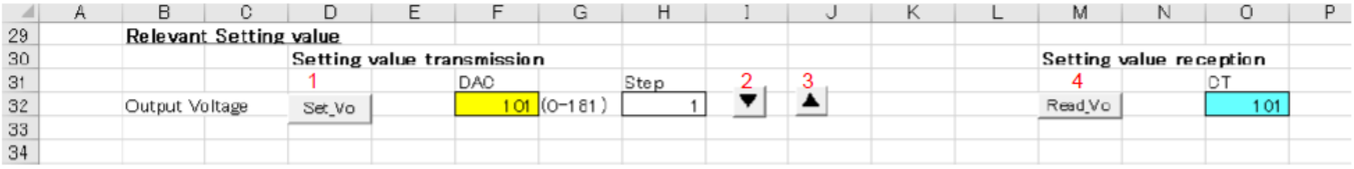
Figure 3-6. TxRxInterface sheet

1. Input CmdGr (command group) in decimal.
2. Input CmdNo (command number) in decimal.
3. Input DATA in decimal. Figure 3-6 is inputting commands to change the D/A converter code in the microcontroller. By inputting CmdGr="4″ and CmdNo= "0", the code value of the microcomputer D/A converters can be changed to the value inputted in DATA. Depending on the command, certain register values in the microcontroller can be received. See Section 4 "Communication Commands List" for the commands that can be communicated with the buck converter EVK.
4. Click, "Send16". When the button is clicked, data is transmitted to the microcontroller, and the value is displayed in the cell of the receiving item below the 19th line when communication is successfully performed. In Figure 3-6, since the setting value is "101", "101" is displayed in J23 cell.

M26 cell displays 6 frames of received data and error messages. An error message is displayed when USB cable is not connected to the USB-UART convert module, when a communication error occurs, or when there is no description of the communication command in RMOS. The description of VBA macro registered in this sheet does not need to be changed because various GUI are created based on VBA macro in TxRxInterface sheet.

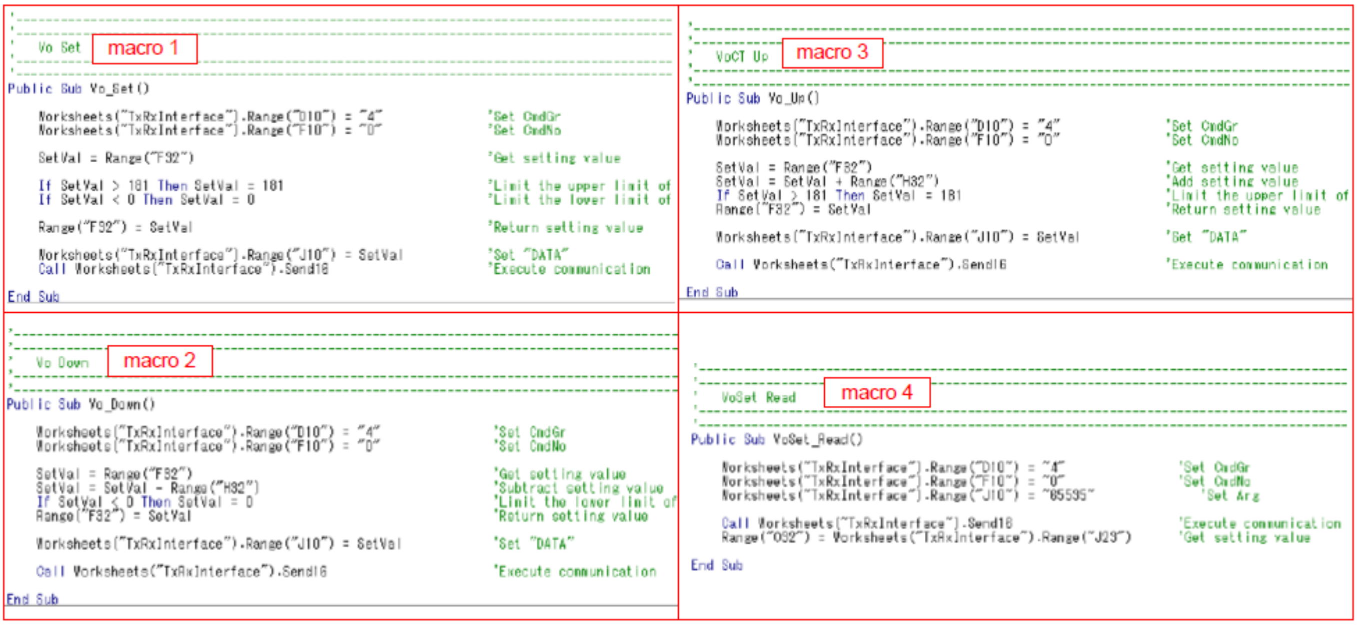
How to use TinyGUI Sheet

This section explains how to use the TinyGUI sheet. This sheet simplifies manual operation on the TxRxInterface sheet so that communication can be performed only by button operation using a VBA macro. Provided Excel file incorporates VBA macros to change the output voltage of the buck converter EVK in the TinyGUI sheet. Figure 3-7 shows the macro button on the TinyGUI sheet. VBA macro is registered in buttons 1 to 4. Figure 3-8 shows the description of VBA macros registered in buttons 1 to 4. This chapter describes an experimental system in which 12V is applied to the input voltage of the buck converter EVK as an example.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
29		Relevant Setting value														
30				Setting value transmission									Setting value reception			
31				1		DAC		Step	2	3			4		DT	
32		Output Voltage		Set_Vo		101	(0-181)	1	▼	▲			Read_Vo		101	
33																
34																

Figure 3-7. Macro button on TinyGUI sheet



macro 1

```
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'   Vo Set
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
Public Sub Vo_Set()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo

    SetVal = Range("F32")                                       'Get setting value

    If SetVal > 181 Then SetVal = 181                           'Limit the upper limit of
    If SetVal < 0 Then SetVal = 0                               'Limit the lower limit of

    Range("F32") = SetVal                                       'Return setting value

    Worksheets("TxRxInterface").Range("J10") = SetVal           'Set "DATA"
    Call Worksheets("TxRxInterface").Send16                     'Execute communication

End Sub
```

macro 3

```
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'   VoCT Up
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
Public Sub Vo_Up()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo

    SetVal = Range("F32")                                       'Get setting value
    SetVal = SetVal + Range("H32")                              'Add setting value
    If SetVal > 181 Then SetVal = 181                           'Limit the upper limit of
    Range("F32") = SetVal                                       'Return setting value

    Worksheets("TxRxInterface").Range("J10") = SetVal           'Set "DATA"

    Call Worksheets("TxRxInterface").Send16                     'Execute communication

End Sub
```

macro 2

```
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'   Vo Down
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
Public Sub Vo_Down()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo

    SetVal = Range("F32")                                       'Get setting value
    SetVal = SetVal - Range("H32")                              'Subtract setting value
    If SetVal < 0 Then SetVal = 0                               'Limit the lower limit of
    Range("F32") = SetVal                                       'Return setting value

    Worksheets("TxRxInterface").Range("J10") = SetVal           'Set "DATA"

    Call Worksheets("TxRxInterface").Send16                     'Execute communication

End Sub
```

macro 4

```
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'   VoSet Read
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
Public Sub VoSet_Read()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo
    Worksheets("TxRxInterface").Range("J10") = "65535"            'Set Arg

    Call Worksheets("TxRxInterface").Send16                     'Execute communication
    Range("O32") = Worksheets("TxRxInterface").Range("J23")     'Get setting value

End Sub
```

Figure 3-8. VBA macros registered in 1 to 4

1. Set_Vo button: Transmits the value of F32 cell to the microcontroller.
2. Down button: Subtracts the code of D/A converter for each number written in H32 cell and transmits the value to the microcontroller.
3. Up button: Adds the code of D/A converter for each number written in H32 cell and transmits the value to the microcontroller.
4. Read_Vo button: The D/A converter code is received from the microcontroller and the received value is displayed in O32 cell.

The output-voltage Vo of the buck converter EVK is determined by the code of D/A converter, and the following equation is established.

$$V_o = \frac{VDD}{256} \times \mathrm{D/A\ code} \times \left(\frac{R_{29} + R_{30} + R_{31}}{R_{31}}\right)$$

- …VDD(microcontroller supply voltage)=5V, R29=51Ω, R30=3.3kΩ, R31=2.2kΩ

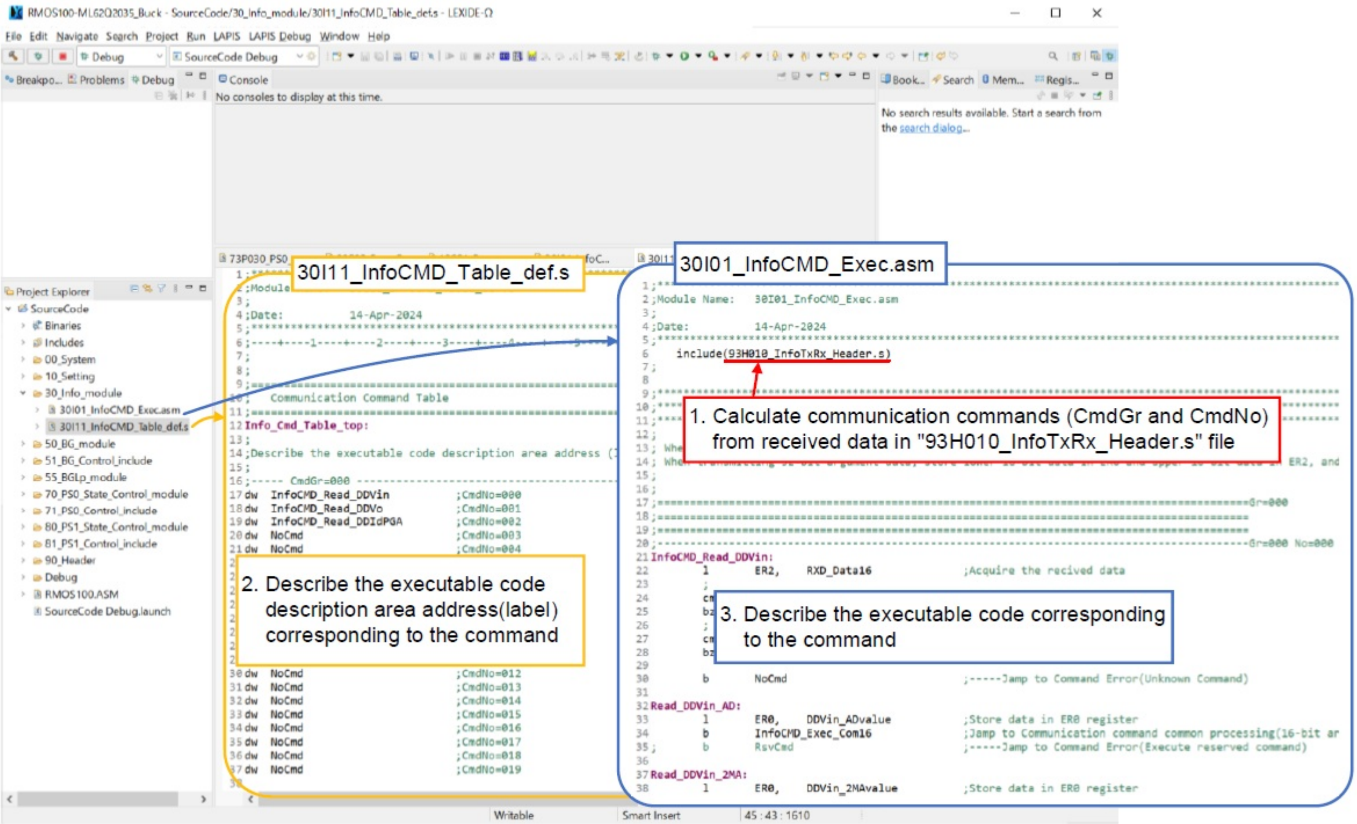
The output voltage Vo of the buck converter EVK can be changed using the setting code calculated from the above equation (Ex. 3.3V="67, 5V="101 and 9V="181). In actual operation, after inputting the value of F32 cell, click button and check that the output voltage of buck converter EVK has changed.

Commands implemented in RMOS

This section describes the program of the communication commands already implemented in RMOS (version "RMOSVer=1.00, OSBuildNo=007, PSFMNo=001, PSFMVer=1.00, PSFMBuildNo=004"). The following two types of commands are described in Section 4 "Communication Commands": Read system commands (commands to read microcontroller values) and Set system commands (commands to write to the MCU). Table 3-1 summarizes the files in "30 Info_module" folder described in this chapter. Figure 3-9 shows the window in "30 Info_module" folder of LEXIDE-Ω.

Table 3-1. Files in the "30_Info_module" folder

No.	File name	Description	Changes
1	30I01_InfoCMD_Exec.asm	Write the program of communication command.	Add a program of additional commands
2	30I11_InfoCMD_Table_def.s	Communication command assignment	Assign labels for additional commands



30I11_InfoCMD_Table_def.s

```
;*****
;Module
;
;Date:          14-Apr-2024
;*******************************************************
;----+----1----+----2----+----3----+----4----+----5---
;
;
;=======================================================
;   Communication Command Table
;=======================================================
Info_Cmd_Table_top:
;
;Describe the executable code description area address (I
;
;----- CmdGr=000 ---------------------------------------
dw  InfoCMD_Read_DDVin          ;CmdNo=000
dw  InfoCMD_Read_DDVo           ;CmdNo=001
dw  InfoCMD_Read_DDIdPGA        ;CmdNo=002
dw  NoCmd                       ;CmdNo=003
dw  NoCmd                       ;CmdNo=004
```

2. Describe the executable code description area address(label) corresponding to the command

```
dw  NoCmd                       ;CmdNo=012
dw  NoCmd                       ;CmdNo=013
dw  NoCmd                       ;CmdNo=014
dw  NoCmd                       ;CmdNo=015
dw  NoCmd                       ;CmdNo=016
dw  NoCmd                       ;CmdNo=017
dw  NoCmd                       ;CmdNo=018
dw  NoCmd                       ;CmdNo=019
```

30I01_InfoCMD_Exec.asm

```
;*******************************************************************************
;Module Name:   30I01_InfoCMD_Exec.asm
;
;Date:          14-Apr-2024
;*******************************************************************************
    include(93H010_InfoTxRx_Header.s)
;

;*******************************************************************************
;*****                                                                 *********
;*****                                                                 *********
;
;  Whe
;  Whe                                                                  ER2, and
;
;
;=============================================================================Gr=000
;==========================================================================
;==========================================================================
;-----------------------------------------------------------------------Gr=000 No=000
InfoCMD_Read_DDVin:
        l       ER2,    RXD_Data16              ;Acquire the recived data
        ;
        cm
        bz
        ;
        cm
        bz

        b       NoCmd                           ;-----Jamp to Command Error(Unknown Command)

Read_DDVin_AD:
        l       ER0,    DDVin_ADvalue           ;Store data in ER0 register
        b       InfoCMD_Exec_Com16              ;Jamp to Communication command common processing(16-bit ar
;       b       RsvCmd                          ;-----Jamp to Command Error(Execute reserved command)

Read_DDVin_2MA:
        l       ER0,    DDVin_2MAvalue          ;Store data in ER0 register
```

1. Calculate communication commands (CmdGr and CmdNo) from received data in "93H010_InfoTxRx_Header.s" file

3. Describe the executable code corresponding to the command

RMOS100-ML62Q2035_Buck - SourceCode/30_Info_module/30I11_InfoCMD_Table_def.s - LEXIDE-Ω
File Edit Navigate Search Project Run LAPIS LAPIS Debug Window Help
Debug
SourceCode Debug
Breakpo... Problems Debug
Console
No consoles to display at this time.
Book... Search Mem... Regis...
No search results available. Start a search from the search dialog...
Project Explorer
SourceCode
Binaries
Includes
00_System
10_Setting
30_Info_module
30I01_InfoCMD_Exec.asm
30I11_InfoCMD_Table_def.s
50_BG_module
51_BG_Control_include
55_BGLp_module
70_PS0_State_Control_module
71_PS0_Control_include
80_PS1_State_Control_module
81_PS1_Control_include
90_Header
Debug
RMOS100.ASM
SourceCode Debug.launch
73P030_PS0
Writable
Smart Insert
45 : 43 : 1610

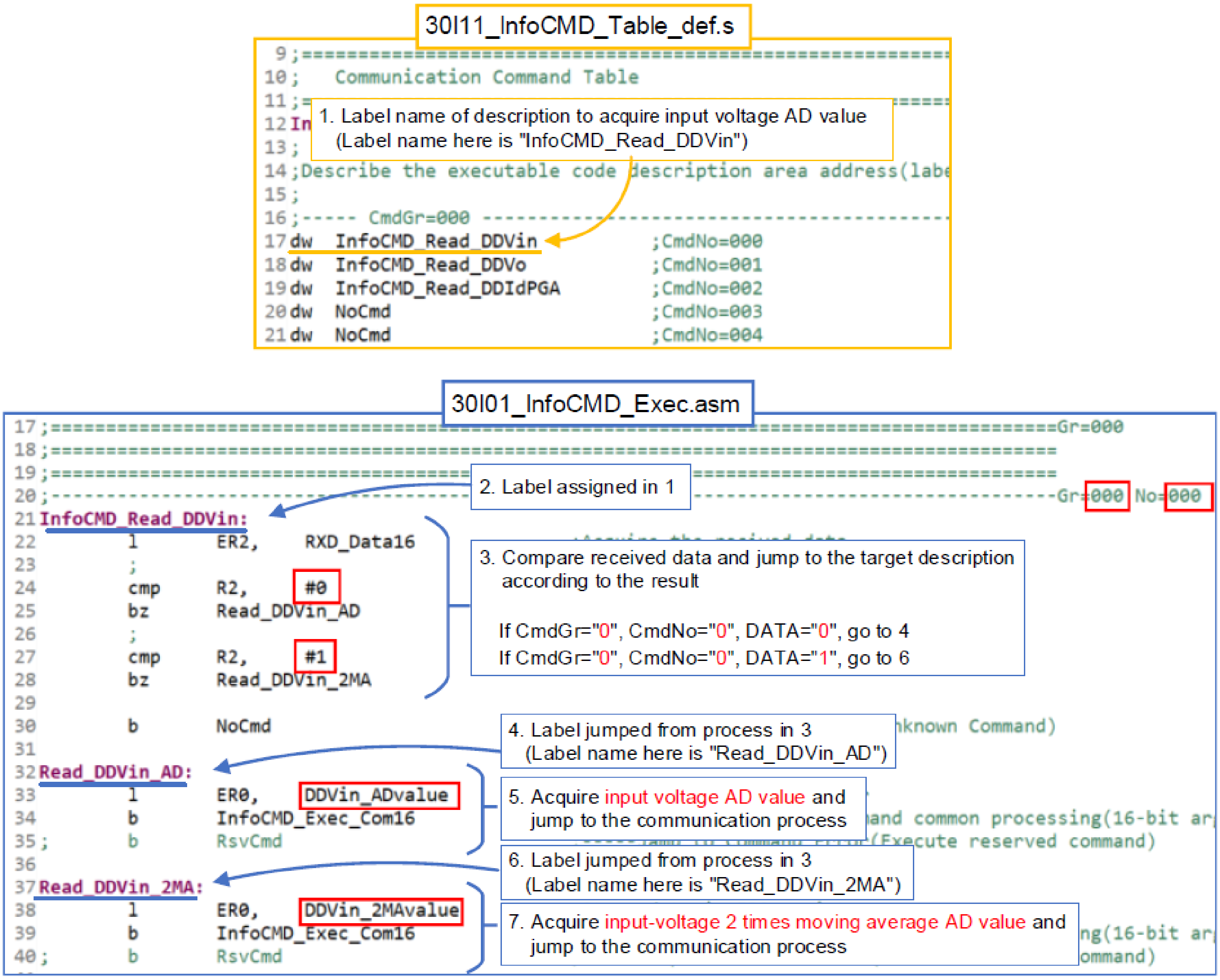
Figure 3-9. Displaying files in "30_Info_module" folder under LEXIDE-Ω

("1 to 3" is the processing of communication data.)

1. TX16 of received data from PC is processed in “93H010_InfoTxRx_Header.s” file. And CmdGr and CmdNo are calculated.
2. Calculated command value indicates the address value of the communication command table in “30I11_InfoCMD_Table_ def.s” file.
3. Jump to the label in “30I01_InfoCMD_Exec.asm” file that matches the read label name in 2 and executes the corresponding program. Note that “93H010_InfoTxRx_Header.s” file has a description that processes the transmitted and received data. Normally, it does not need to be edited. Read and Set commands described in “30I11_InfoCMD_Table_def.s” file and “30I01_InfoCMD_Exec.asm” file are described in (1), (2) below.

Read Command

Figure 3-10 shows the descriptions and operations of Read command. The following programs describe how to read an input voltage AD value or 2 times moving average value (when CmdGr = “0” and CmdNo = “0”).



30I11_InfoCMD_Table_def.s

```
;==============================================================
;    Communication Command Table
;=                                                        =====
In
;
;Describe the executable code description area address(labe
;
;----- CmdGr=000 -------------------------------------------
dw   InfoCMD_Read_DDVin                   ;CmdNo=000
dw   InfoCMD_Read_DDVo                    ;CmdNo=001
dw   InfoCMD_Read_DDIdPGA                 ;CmdNo=002
dw   NoCmd                                ;CmdNo=003
dw   NoCmd                                ;CmdNo=004
```

1. Label name of description to acquire input voltage AD value (Label name here is "InfoCMD_Read_DDVin")

30I01_InfoCMD_Exec.asm

```
;==========================================================================Gr=000
;========================================================================
;========================================================================
;-------------------------------------------------------------Gr=000 No=000
InfoCMD_Read_DDVin:
        l       ER2,    RXD_Data16
        ;
        cmp     R2,     #0
        bz      Read_DDVin_AD
        ;
        cmp     R2,     #1
        bz      Read_DDVin_2MA

        b       NoCmd                         nknown Command)

Read_DDVin_AD:
        l       ER0,    DDVin_ADvalue
        b       InfoCMD_Exec_Com16            and common processing(16-bit ar
;       b       RsvCmd                         (Execute reserved command)

Read_DDVin_2MA:
        l       ER0,    DDVin_2MAvalue
        b       InfoCMD_Exec_Com16            ng(16-bit ar
;       b       RsvCmd                        ommand)
```

2. Label assigned in 1

3. Compare received data and jump to the target description according to the result

If CmdGr="0", CmdNo="0", DATA="0", go to 4
If CmdGr="0", CmdNo="0", DATA="1", go to 6

4. Label jumped from process in 3 (Label name here is "Read_DDVin_AD")

5. Acquire input voltage AD value and jump to the communication process

6. Label jumped from process in 3 (Label name here is "Read_DDVin_2MA")

7. Acquire input-voltage 2 times moving average AD value and jump to the communication process

Figure 3-10. Description and operation of read command

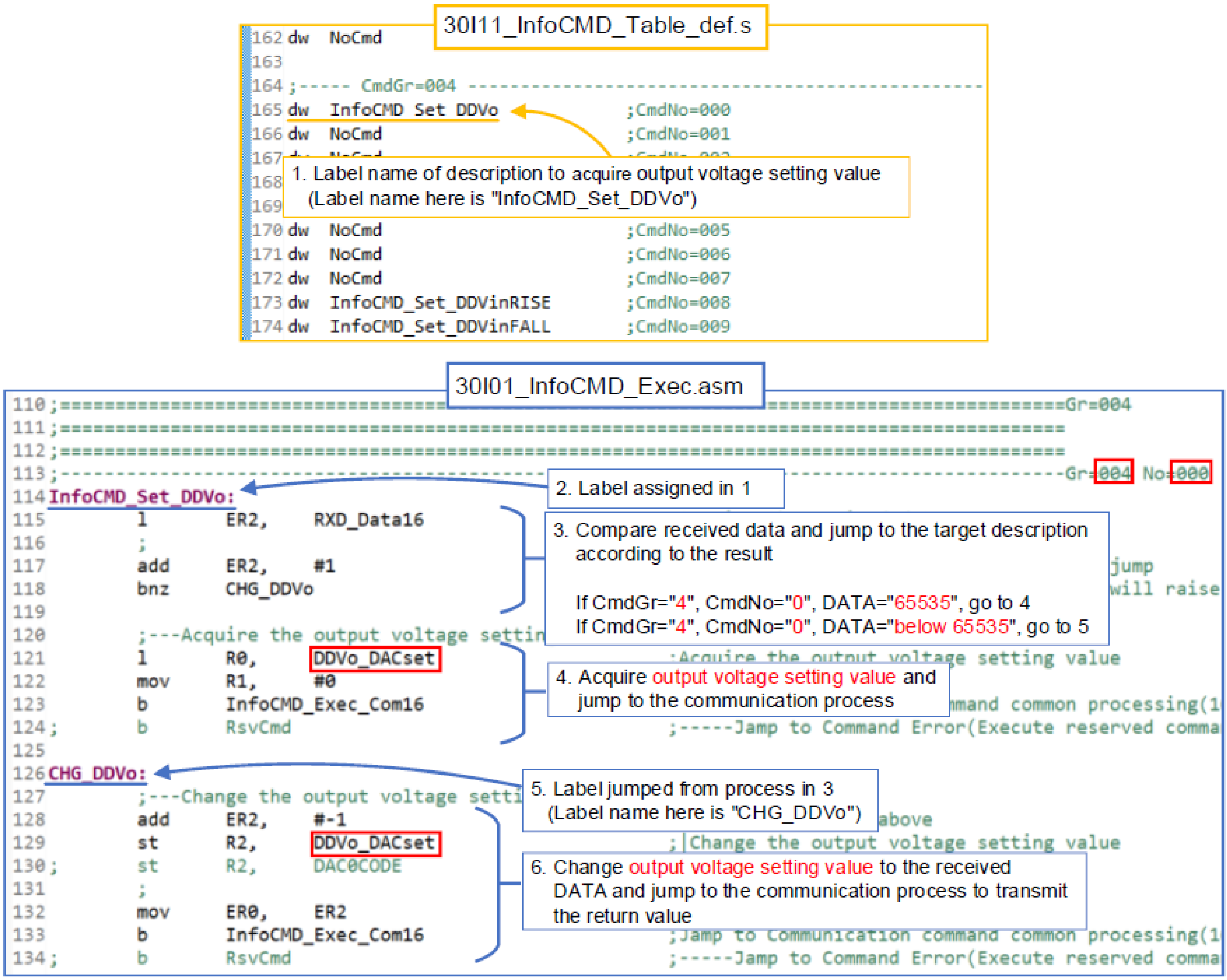
1. The beginning label name of the description to acquire the input voltage AD value is assigned in “30I11_InfoCMD_Table_ def.s” file. The received data is processed in “93H010_InfoTxRx_Header.s” file and jumped to “InfoCMD_Read_DDVin” label when CmdGr = “0” and CmdNo = “0”.
2. The label name assigned in 1 is described, and the description below the label is executed.
3. Compare data received from the PC and jump target descriptions accordingly.
   - If CmdGr = “0”, CmdNo = “0”, DATA = “0”, go to 4
   - If CmdGr = “0”, CmdNo = “0”, DATA = “1”, go to 6
4. The label jumped from the processing of 3 is described. Then, the label name is “Read_DDVin_AD”.
5. Acquire “DDVin_ADvalue”(input voltage AD value) and jump to the communication process to transmit the return value to PC.
6. The label jumped from the processing of 3 is described. Then, the label name is

"Read_DDVin_2MA".

7. Acquire "DDVin_2MAvalue"(input voltage 2 times moving average AD value) and jump to the communication process to transmit the return value to PC.

Set Command

Figure 3-11 shows the descriptions and operations of the Set command. The following commands describe how to acquire and change the output voltage (when CmdGr = "4" and CmdNo = "0").



30I11_InfoCMD_Table_def.s

```
dw   NoCmd

;----- CmdGr=004 ---------------------------------------------
dw   InfoCMD_Set_DDVo                ;CmdNo=000
dw   NoCmd                           ;CmdNo=001
dw   NoCmd                           ;CmdNo=002



dw   NoCmd                           ;CmdNo=005
dw   NoCmd                           ;CmdNo=006
dw   NoCmd                           ;CmdNo=007
dw   InfoCMD_Set_DDVinRISE           ;CmdNo=008
dw   InfoCMD_Set_DDVinFALL           ;CmdNo=009
```

1. Label name of description to acquire output voltage setting value (Label name here is "InfoCMD_Set_DDVo")

30I01_InfoCMD_Exec.asm

```
;============================================================================Gr=004
;==========================================================================
;==========================================================================
;----------------------------------------------------------------Gr=004 No=000
InfoCMD_Set_DDVo:
        l       ER2,    RXD_Data16
        ;
        add     ER2,    #1                                                  jump
        bnz     CHG_DDVo                                                    will raise

        ;---Acquire the output voltage settin
        l       R0,     DDVo_DACset             ;Acquire the output voltage setting value
        mov     R1,     #0
        b       InfoCMD_Exec_Com16                  mmand common processing(1
;       b       RsvCmd                          ;-----Jamp to Command Error(Execute reserved comma

CHG_DDVo:
        ;---Change the output voltage setti
        add     ER2,    #-1                     above
        st      R2,     DDVo_DACset             ;|Change the output voltage setting value
;       st      R2,     DAC0CODE
        ;
        mov     ER0,    ER2
        b       InfoCMD_Exec_Com16              ;Jamp to Communication command common processing(1
;       b       RsvCmd                          ;-----Jamp to Command Error(Execute reserved comma
```

2. Label assigned in 1

3. Compare received data and jump to the target description according to the result

If CmdGr="4", CmdNo="0", DATA="65535", go to 4
If CmdGr="4", CmdNo="0", DATA="below 65535", go to 5

4. Acquire output voltage setting value and jump to the communication process

5. Label jumped from process in 3 (Label name here is "CHG_DDVo")

6. Change output voltage setting value to the received DATA and jump to the communication process to transmit the return value

Figure 3-11. Description and operation of set command

1. The beginning label name of the description that acquires or changes the output voltage is assigned in "30I11_InfoCMD_Table _def.s" file. The data is processed in "93H010_InfoTxRx_Header.s" file and jumped to "InfoCMD_Set_DDVo" label when CmdGr = "4" and CmdNo = "0".
2. The label name assigned in 1 is described, and the description below the label is executed.
3. Compare data received from PC and jump target descriptions accordingly.

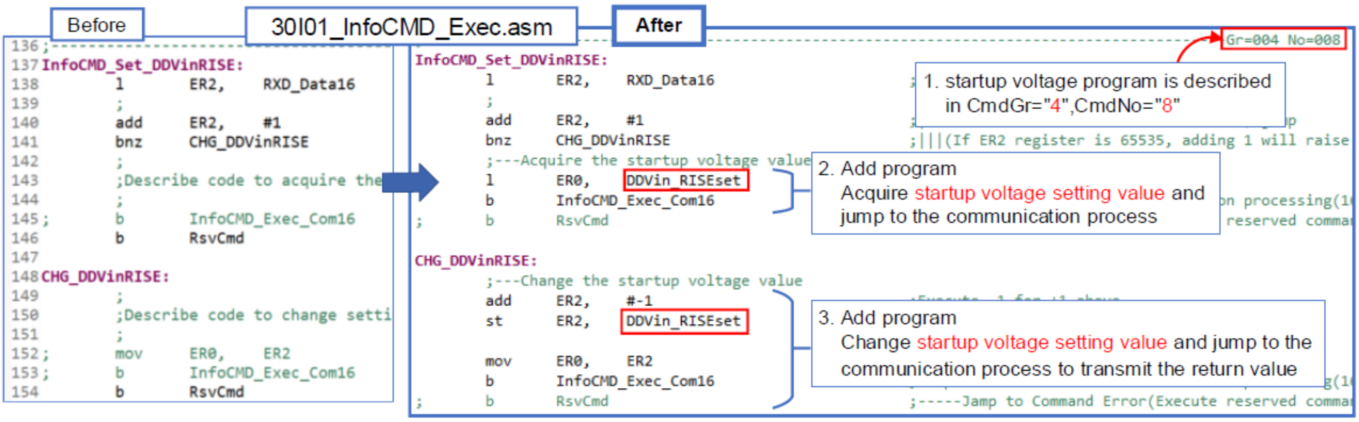
- If CmdGr = “4”, CmdNo = “0”, DATA = “65535”, go to 4
- If CmdGr = “4”, CmdNo = “0”, DATA = “below 65535”, go to 5

4. Acquire “DDVo_DACset”(output-voltage setting value) and jump to the communication process to transmit the return value to PC. This description is the same as the Read command. When DATA=”65535” is inputted, “DDVo_DACset” is read from the microcontroller.
5. The label jumped from the processing of 3 is described. Then, the label name is “CHG_DDVo”.
6. Change “DDVo_DACset” to the value of DATA and jump to the communication process to transmit the return value to PC. The value of “DDVo_DACset” in the microcontroller can be changed by this description.

Please use TxRxInterface sheet of Excel file to check the operation of the communication commands (Refer to 3-3 “How to use TinyGUI sheet”). Even if DATA=”65535″ is transmitted immediately after Excel is started, the initialization setting is “101”, so the reception status remains unchanged.

How to Add Set Command

In the Set commands in Chapter 4 “Communication Commands List”, some of the commands are not described except for CmdGr=”4″ and CmdNo=”0″. Therefore, it is necessary to add them. Figure 3-12 shows an example of adding a communication command to change the startup voltage setting value in steps 1 to 3. To use a command other than this one, write the corresponding state variable name from Table 4-1 in the red box.



```
;-----------------------------------
InfoCMD_Set_DDVinRISE:
        l       ER2,    RXD_Data16
        ;
        add     ER2,    #1
        bnz     CHG_DDVinRISE
        ;
        ;Describe code to acquire the
        ;
;       b       InfoCMD_Exec_Com16
        b       RsvCmd

CHG_DDVinRISE:
        ;
        ;Describe code to change setti
        ;
;       mov     ER0,    ER2
;       b       InfoCMD_Exec_Com16
        b       RsvCmd
```

```
;------------------------------------------------------------ Gr=004 No=008
InfoCMD_Set_DDVinRISE:
        l       ER2,    RXD_Data16
        ;
        add     ER2,    #1                                ;|||(If ER2 register is 65535, adding 1 will raise
        bnz     CHG_DDVinRISE
        ;---Acquire the startup voltage value
        l       ER0,    DDVin_RISEset                     on processing(1
        b       InfoCMD_Exec_Com16                        reserved comma
;       b       RsvCmd

CHG_DDVinRISE:
        ;---Change the startup voltage value
        add     ER2,    #-1
        st      ER2,    DDVin_RISEset

        mov     ER0,    ER2
        b       InfoCMD_Exec_Com16
;       b       RsvCmd                                    ;-----Jamp to Command Error(Execute reserved comma
```

1. startup voltage program is described in CmdGr="4",CmdNo="8"

2. Add program
Acquire startup voltage setting value and jump to the communication process

3. Add program
Change startup voltage setting value and jump to the communication process to transmit the return value

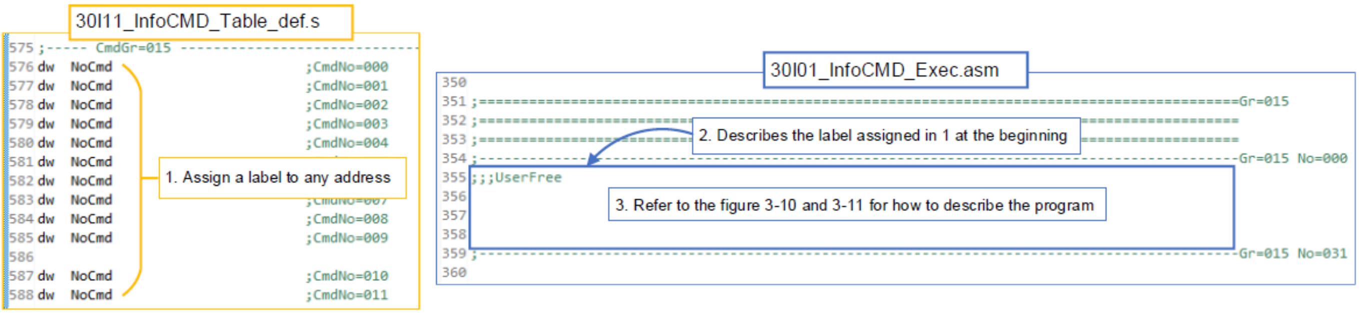
Figure 3-12. How to expand communication command of start voltage

1. The program to set the starting voltage value “DDVin_RISEset” is written in CmdGr=”4″ and CmdNo=”8″.

2. Acquire “DDVin_RISEset”(startup voltage setting value) and jump to the communication process to transmit the return value to PC.
3. Change “DDVin_RISEset” to the value of DATA and jump to the communication process to transmit the return value to PC.

How to add command in UserFree area

This section describes how to read and change state variables not listed in Chapter 4, “Communication Commands List”. Follow the steps 1 to 3 below to add more commands. Program area that can be freely described by user are provided in CmdGr=”15″.



30I11_InfoCMD_Table_def.s

```
;----- CmdGr=015 ------------------------------
dw   NoCmd                           ;CmdNo=000
dw   NoCmd                           ;CmdNo=001
dw   NoCmd                           ;CmdNo=002
dw   NoCmd                           ;CmdNo=003
dw   NoCmd                           ;CmdNo=004
dw   NoCmd
dw   NoCmd
dw   NoCmd                           ;CmdNo=007
dw   NoCmd                           ;CmdNo=008
dw   NoCmd                           ;CmdNo=009

dw   NoCmd                           ;CmdNo=010
dw   NoCmd                           ;CmdNo=011
```

1. Assign a label to any address

30I01_InfoCMD_Exec.asm

```

;==========================================================================Gr=015
;==========================================================================
;==========================================================================
;--------------------------------------------------------------------------Gr=015 No=000
;;;UserFree



;--------------------------------------------------------------------------Gr=015 No=031

```

2. Describes the label assigned in 1 at the beginning

3. Refer to the figure 3-10 and 3-11 for how to describe the program

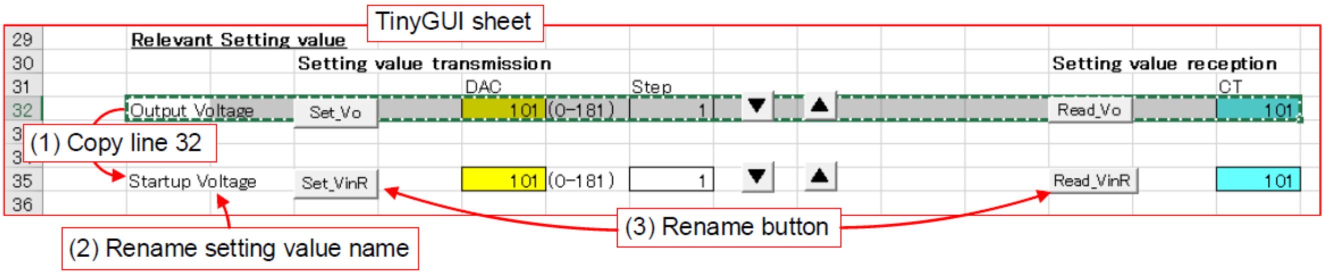
Figure 3-13. How to expand communication command of startup voltage

1. Assign labels to any CmdNo.
2. Describe the label assigned in 1 at the beginning of the description.
3. Describe the command processing. Refer to Figures 3-10 and 3-11 for how to describe the program.

How to add a GUI

This chapter describes how to operate the TxRxInterface sheet using the GUI. This makes debugging the power supply easier, as power supply parameters can be easily changed and acquired. GUI can create in TinyGUI sheet of Excel file. User can add GUI as shown in steps 1 to 5 below. In this chapter, GUI is created by referring to Chapter 3-5 (1) “How to add Set command” and changing the startup voltage (CmdGr=”4″ and CmdNo=”8″).

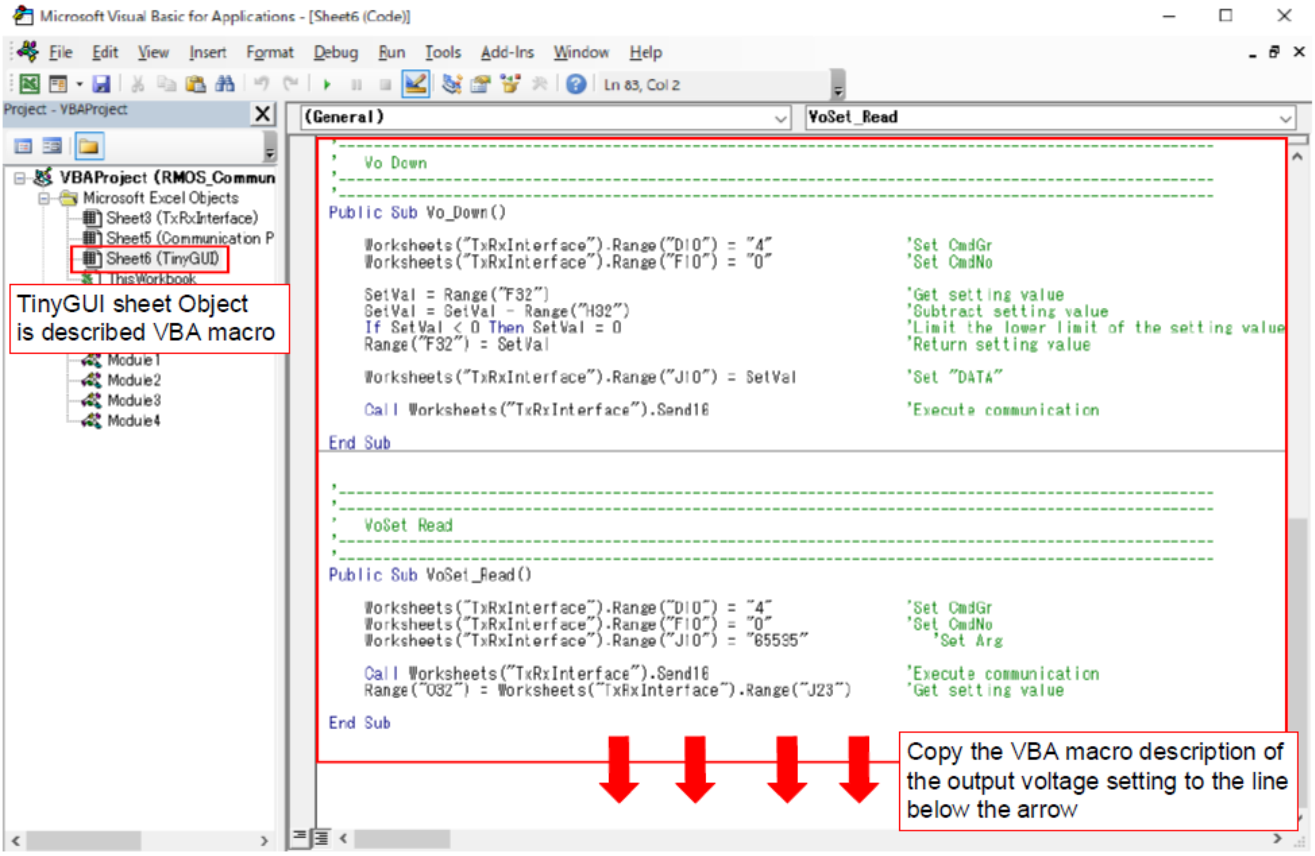
1. Copy macro buttons and cells in a sheet. And rename macro buttons.



29	Relevant Setting value									
30		Setting value transmission							Setting value reception	
31			DAC		Step					CT
32	Output Voltage	Set_Vo	101	(0-181)	1	▼	▲	Read_Vo		101
35	Startup Voltage	Set_VinR	101	(0-181)	1	▼	▲	Read_VinR		101
36										

Figure 3-14. Step on TinyGUI sheet

2. Copy the pre-registered VBA macro (Figures 3-8) to the line below the arrow.

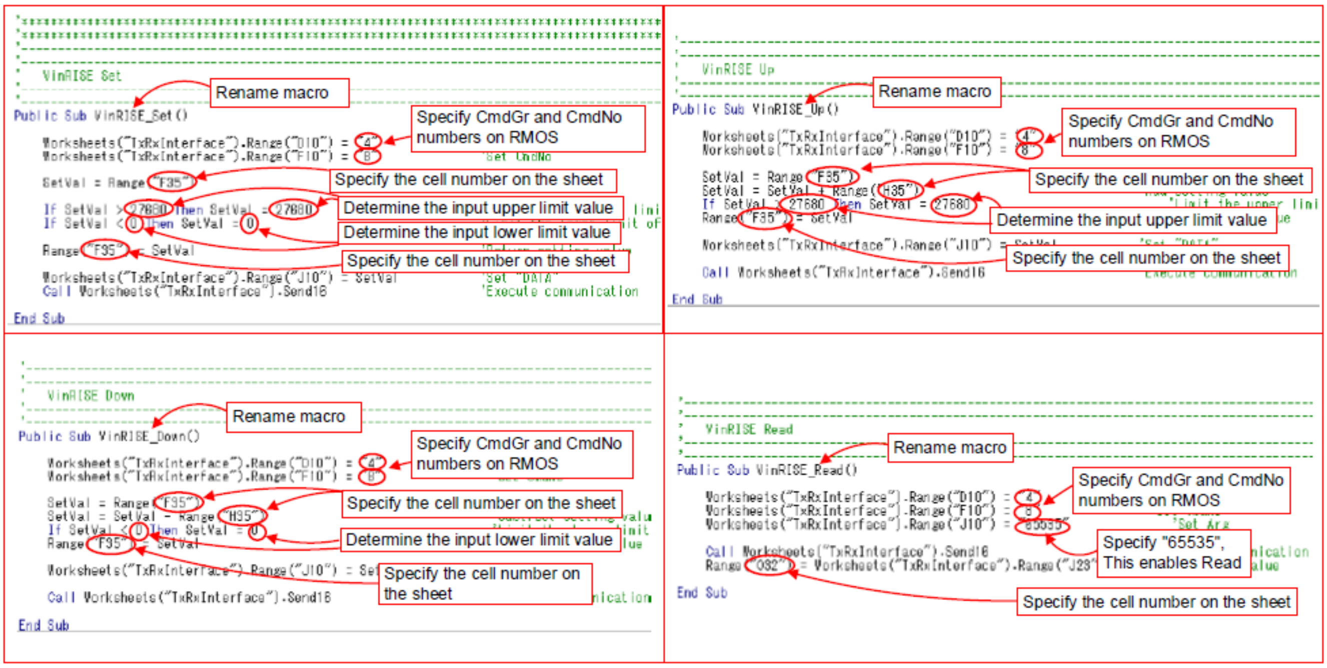


```
'------------------------------------------------------------------------------
'   Vo Down
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
Public Sub Vo_Down()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo

    SetVal = Range("F32")                                       'Get setting value
    SetVal = SetVal - Range("H32")                              'Subtract setting value
    If SetVal < 0 Then SetVal = 0                               'Limit the lower limit of the setting value
    Range("F32") = SetVal                                       'Return setting value

    Worksheets("TxRxInterface").Range("J10") = SetVal           'Set "DATA"

    Call Worksheets("TxRxInterface").Send16                     'Execute communication

End Sub

'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
'   VoSet Read
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
Public Sub VoSet_Read()

    Worksheets("TxRxInterface").Range("D10") = "4"              'Set CmdGr
    Worksheets("TxRxInterface").Range("F10") = "0"              'Set CmdNo
    Worksheets("TxRxInterface").Range("J10") = "65535"            'Set Arg

    Call Worksheets("TxRxInterface").Send16                     'Execute communication
    Range("O32") = Worksheets("TxRxInterface").Range("J23")     'Get setting value

End Sub
```

Copy the VBA macro description of the output voltage setting to the line below the arrow

Figure 3-15. Steps on Excel VBA window

3. VBA macro is rewritten according to the description on RMOS or the cell copied in Step 1.



```
'-----------------------------------------------------------
'   VinRISE Set
'-----------------------------------------------------------
Public Sub VinRISE_Set()

    Worksheets("TxRxInterface").Range("D10") = "4"
    Worksheets("TxRxInterface").Range("F10") = "8"               'Set CmdNo

    SetVal = Range("F35")

    If SetVal > 27680 Then SetVal = 27680
    If SetVal < 0 Then SetVal = 0

    Range("F35") = SetVal

    Worksheets("TxRxInterface").Range("J10") = SetVal            'Set "DATA"
    Call Worksheets("TxRxInterface").Send16                      'Execute communication

End Sub
```

Rename macro

Specify CmdGr and CmdNo numbers on RMOS

Specify the cell number on the sheet

Determine the input upper limit value

Determine the input lower limit value

Specify the cell number on the sheet

```
'-----------------------------------------------------------
'   VinRISE Up
'-----------------------------------------------------------
Public Sub VinRISE_Up()

    Worksheets("TxRxInterface").Range("D10") = "4"
    Worksheets("TxRxInterface").Range("F10") = "8"

    SetVal = Range("F35")
    SetVal = SetVal + Range("H35")
    If SetVal > 27680 Then SetVal = 27680                        'Limit the upper limi
    Range("F35") = SetVal

    Worksheets("TxRxInterface").Range("J10") = SetVal            'Set "DATA"

    Call Worksheets("TxRxInterface").Send16

End Sub
```

Rename macro

Specify CmdGr and CmdNo numbers on RMOS

Specify the cell number on the sheet

Determine the input upper limit value

Specify the cell number on the sheet

```
'-----------------------------------------------------------
'   VinRISE Down
'-----------------------------------------------------------
Public Sub VinRISE_Down()

    Worksheets("TxRxInterface").Range("D10") = "4"
    Worksheets("TxRxInterface").Range("F10") = "8"

    SetVal = Range("F35")
    SetVal = SetVal - Range("H35")
    If SetVal < 0 Then SetVal = 0
    Range("F35") = SetVal

    Worksheets("TxRxInterface").Range("J10") = SetVal

    Call Worksheets("TxRxInterface").Send16

End Sub
```

Rename macro

Specify CmdGr and CmdNo numbers on RMOS

Specify the cell number on the sheet

Determine the input lower limit value

Specify the cell number on the sheet

```
'-----------------------------------------------------------
'   VinRISE Read
'-----------------------------------------------------------
Public Sub VinRISE_Read()

    Worksheets("TxRxInterface").Range("D10") = "4"
    Worksheets("TxRxInterface").Range("F10") = "8"
    Worksheets("TxRxInterface").Range("J10") = "65535"           'Set Arg

    Call Worksheets("TxRxInterface").Send16
    Range("O32") = Worksheets("TxRxInterface").Range("J23"

End Sub
```

Rename macro

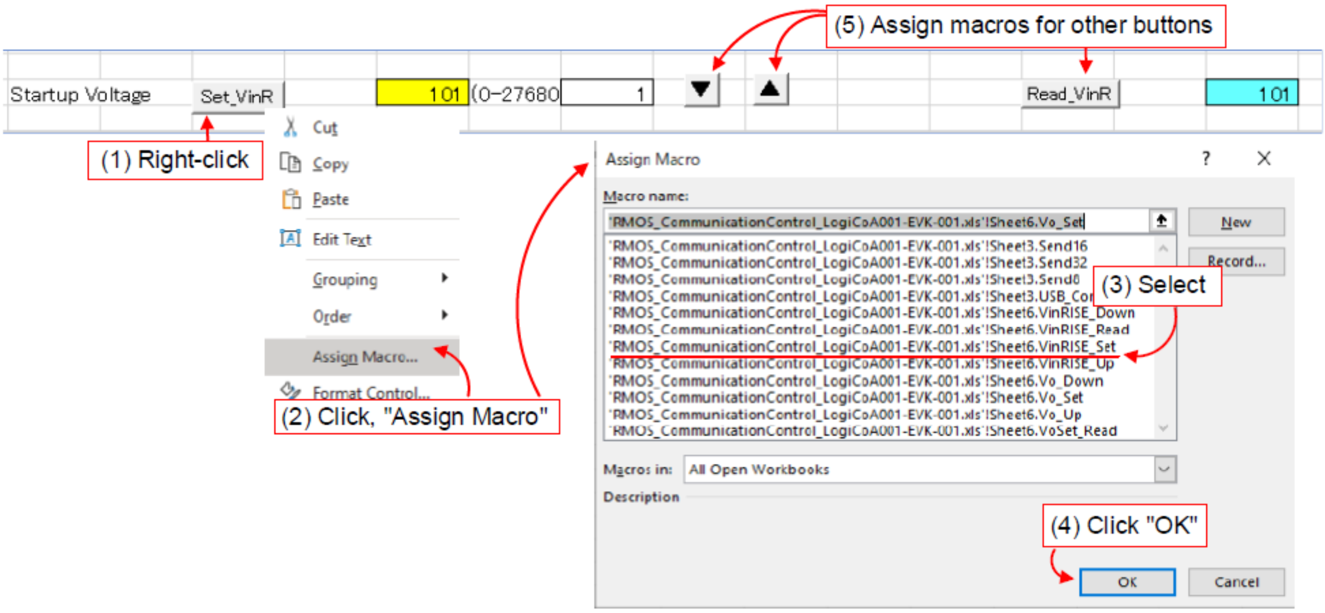
Specify CmdGr and CmdNo numbers on RMOS

Specify "65535", This enables Read

Specify the cell number on the sheet

Figure 3-16. Excel VBA macro setting

4. Register the macro name added with VBA to the macro button. (Steps of (1) to (5))



(5) Assign macros for other buttons

(1) Right-click

(2) Click, "Assign Macro"

Macro name:

'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.Vo_Set

'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet3.Send16
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet3.Send32
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet3.Send8
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet3.USB_Cor
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.VinRISE_Down
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.VinRISE_Read
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.VinRISE_Set
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.VinRISE_Up
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.Vo_Down
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.Vo_Set
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.Vo_Up
'RMOS_CommunicationControl_LogiCoA001-EVK-001.xls'!Sheet6.VoSet_Read

(3) Select

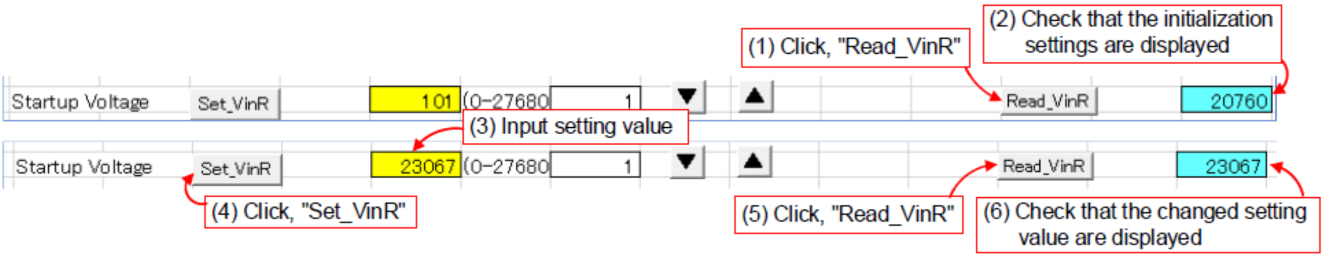
Macros in: All Open Workbooks

Description

(4) Click "OK"

Figure 3-17. Steps for registering VBA macro

5. Check the operation of the created GUI. (Steps of (1) to (6))



(1) Click, "Read_VinR"

(2) Check that the initialization settings are displayed

(3) Input setting value

(4) Click, "Set_VinR"

(5) Click, "Read_VinR"

(6) Check that the changed setting value are displayed

Figure 3-18. Step for checking GUI operation

User can easily create a GUI for communication using the above steps. In the GUI shown in step 5, the startup voltage is input as AD value, but it can also be input as a voltage value by the function of Excel file. For the calculation method for converting startup voltage AD value to the voltage value, refer to the User’s Guide [2] and Operating Manual [3] of the buck converter. Parameters such as stop voltage, dead time, and OCP (over current protection) can also be added to the GUI in the same step as the startup voltage.

Communication Commands List

Table 4-1 contains a list of CmdGr and CmdNo functions. In RMOS (version “RMOSVer=1.00, OSBuildNo=007, PSFMNo=001, PSFMVer=1.00, PSFMBuildNo=004”) in this manual, CmdGr of 1 to 15 can be used, and CmdNo of 1 to 31 can be used. Please note that the command’s configuration in the table 4-1 may change due to future RMOS upgrades.

Table 4-1. Communication Commands List

Read commands

Cmd Gr	Cmd No	DATA	Command label name	State variable name	Command description
0	0	0	InfoCMD_Read_DDVin	DDVin_ADvalue	Input voltage AD value
		1		DDVin_2MAvalue	2 times moving average AD
	1	0	InfoCMD_Read_DDVo	DDVo_ADvalue	Output voltage AD value
		1		DDVo_2MAvalue	2 times moving average AD
		0		DDIdPGA_ADvalue	Drain current AD value

	2	1	InfoCMD_Read_DDIdPGA	DDIdPGA_2MAvalue	2 times moving average AD
		2		DDIdPGA_8MAvalue	8 times moving average AD

Set commands

Cmd Gr	Cmd No	DATA	Command label name	State variable name	Command description
4	0	Value	InfoCMD_Set_DDVo	DDVo_DACset	Output Voltage setting
	8	Value	InfoCMD_Set_DDVinRISE	DDVin_RISEset	Startup voltage setting
	9	Value	InfoCMD_Set_DDVinFALL	DDVin_FALLset	Stop voltage setting
5	0	Value	InfoCMD_DD0_Fsw	Fsw_CTset	Switching Frequency setting
	16	Value	InfoCMD_DD0_TonMax	dmax_CTset	Switching device maximum duty setting
6	0	Value	InfoCMD_DD0_OCP	DDOCP_Ioset	OCP setting
7	0	Value	InfoCMD_DD0_OVP	DDOVP_VoADset	OVP setting
	16	Value	InfoCMD_DD0_LVP	DDLVP_VoADset	LVP setting
	0	Value	InfoCMD_DD0_DeadTime0	DTimeHoffLon_CTset	Dead time 0 count value setting

11	1	value	InfoCMD_DD0_DeadTime1	DTimeLoffHon_CTset	Dead time 1 count value setting

Please note that CmdGr and CmdNo that are not specified above are reserved areas and may add programs in future RMOS upgrades or EVK releases.

UserFree commands

CmdGr	CmdNo	DATA	Command label name	State variable name	Command's description
15	0	–	Can be freely described	–	–
	31	–	–	–	–

References

1. 66AN147E, Rev.001, Operating system for switching power control MCU "RMOS"
2. 66UG090E, Rev.001, Synchronous Buck DCDC Converter Evaluation Board LogiCoA001-EVK-001
3. 66AN153E, Rev.001, Analog-Digital Hybrid Control Power Supply Synchronous Buck DCDC Converter Operating Instructions

Revision History

Date	Revision Number	Description
5. June. 2024	001	Initial release

Notice

1. The information contained in this document is intended to introduce ROHM Group (hereafter referred to as ROHM) products. When using ROHM products, please verify the latest specifications
   or datasheets before use.
2. ROHM products are designed and manufactured for use in general electronic equipment and
   applications (such as Audio Visual equipment, Office Automation equipment, telecommunication equipment, home appliances, amusement devices, etc.) or specified in the datasheets. Therefore,
   please contact the ROHM sales representative before using ROHM products in equipment or devices requiring extremely high reliability and whose failure or malfunction may cause danger or injury to human life or body or other serious damage (such as medical equipment, transportation, traffic, aircraft, spacecraft, nuclear power controllers, fuel control, automotive equipment including
   car accessories, etc. hereafter referred to as Specific Applications). Unless otherwise agreed in writing by ROHM in advance, ROHM shall not be in any way responsible or liable for any damages,
   expenses, or losses incurred by you or third parties arising from the use of ROHM Products for Specific Applications.
3. Electronic components, including semiconductors, can fail or malfunction at a certain rate. Please be sure to implement, at your own responsibilities, adequate safety measures, including but not limited to fail-safe design against physical injury, and

damage to any property, which a failure or malfunction of products may cause.

4. The information contained in this document, including application circuit examples and their constants, is intended to explain the standard operation and usage of ROHM products, and is not intended to guarantee, either explicitly or implicitly, the operation of the product in the actual equipment it will be used. As a result, you are solely responsible for it, and you must exercise your own independent verification and judgment in the use of such information contained in this document. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties arising from the use of such information.
5. When exporting ROHM products or technologies described in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, such as the Foreign Exchange and Foreign Trade Act and the US Export Administration
6. Regulations, and follow the necessary procedures in accordance with these provisions.
7. The technical information and data described in this document, including typical application circuits, are examples only and are not intended to guarantee to be free from infringement of third parties intellectual property or other rights. ROHM does not grant any license, express or implied, to implement, use, or exploit any intellectual property or other rights owned or controlled by ROHM or any third parties with respect to the information contained herein.
8. No part of this document may be reprinted or reproduced in any form by any means without the prior written consent of ROHM.
9. All information contained in this document is current as of the date of publication and subject to change without notice. Before purchasing or using ROHM products, please confirm the latest information with the ROHM sales representative. ROHM does not warrant that the information contained herein is error-free. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties resulting from errors contained in this document.

Thank you for your accessing ROHM product information. More detailed product information and catalogs are available, Please contact us.

ROHM Customer Support System

- https://www.rohm.com/contactus
- www.rohm.com
- 2023 ROHM Co., Ltd. All rights reserved.

FAQs

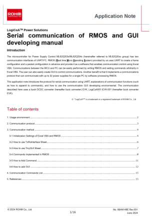
Q: What is the purpose of the RMOS project file?

A: The RMOS project file is used to read into LEXIDE for programming and communication control.

Q: How many frames are there per packet in the communication protocol?

A: The communication protocol consists of 6 frames per packet.

Documents / Resources

 Application Note LogiCoA™ Power Solutions Serial communication of RMOS and GUI developing manual Introduction The microcontroller for Power Supply Control ML62Q203x/ML62Q204x (hereinafter referred to ML62Q20xx group) has two communication interfaces of UART/I²C. RMOS (Real time Micro Operating System) provided by us uses UART to create a frame configuration and a packet configuration in advance and provide it as a software that enables communication control using Excel VBA. Communications between the MCU and PC can be easily performed by writing RMOS and setting commands arbitrarily in Excel VBA. The user can also easily create GUI to control communications. Another benefit is that it implements a communications protocol that can communicate with up to 32 power supplies for a single PC by software processing RMOS. This application note introduces the protocol for serial communication using UART, explanations of communication functions (such as how to append to commands), and how to use the communication GUI developing environmental. The communication described here uses a buck DCDC converter (hereafter buck converter) EVK, LogiCoA001-EVK-001 (hereafter buck converter EVK). © "LogiCoA™" is a trademark or a registered trademark of ROHM Co., Ltd. Table of contents 1. Usage environment..........2 2. Communication protocol..........3 3. Communication method..........4 3-1 Initialization Settings of Excel VBA and RMOS..........4 3-2 How to use TxRxInterface Sheet..........6 3-3 How to use TinyGUI Sheet..........7 3-4 Commands implemented in RMOS..........8 3-5 How to Add Command..........11 3-6 How to add GUI..........12 4. Communication Commands List..........15 5. References..........15 © 2024 ROHM Co., Ltd. 1/16 No. 68AN149E Rev.001 June 2024	ROHM ML62Q203x LogiCoA RMOS Microcontroller for Power Supply Control [pdf] User Guide ML62Q203x, ML62Q203x LogiCoA RMOS Microcontroller for Power Supply Control, LogiCoA RMOS Microcontroller for Power Supply Control, RMOS Microcontroller for Power Supply Control, Microcontroller for Power Supply Control, Power Supply Control

References

- User Manual

ROHM

LogiCoA RMOS Microcontroller for Power Supply Control, Microcontroller for Power Supply Control, ML62Q203x, ML62Q203x LogiCoA RMOS Microcontroller for Power Supply Control, Power Supply Control, RMOS Microcontroller for Power Supply Control, ROHM

Leave a comment

Your email address will not be published. Required fields are marked *

Comment *

Name

Email

Website

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search:

e.g. whirlpool wrf535swhz

Search

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.